HybridTrak: Adding Full-Body Tracking to VR Using an Off-the-Shelf Webcam

Jackie (Junrui) Yang
jackiey@stanford.edu
Stanford University
Stanford, CA, USA

Tuochao Chen
ctc1998@uw.edu
University Of Washington
Seattle, WA, USA

Fang Qin
fangq@stanford.edu
Stanford University
Stanford, CA, USA

Monica S. Lam
lam@cs.stanford.edu
Stanford University
Stanford, CA, USA

James A. Landay
landay@stanford.edu
Stanford University
Stanford, CA, USA

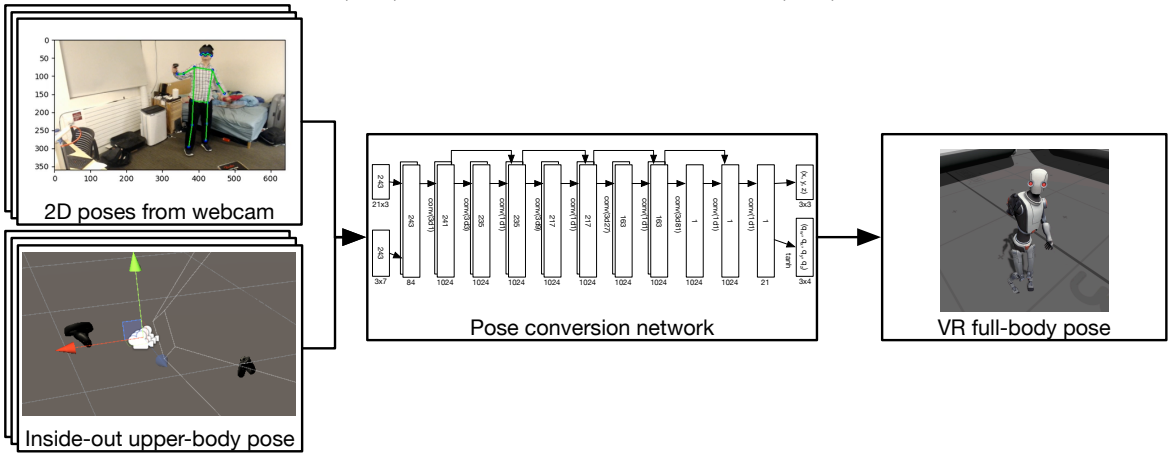

Figure 1: HybridTrak is a VR full-body tracking solution that augments inside-out upper body tracking with a single uncalibrated webcam for lower body tracking. This approach provides accurate lower-body tracking, which normally requires a cumbersome outside-in VR setup, with similar convenience to inside-out tracking. We developed a novel full-neural solution that combines estimated 2D poses from the webcam and upper-body positions and orientations from the VR headset to produce 3D poses of the user in VR coordinates. By emulating virtual devices, HybridTrak is compatible with current VR applications that support full-body tracking on SteamVR.

ABSTRACT

Full-body tracking in virtual reality improves presence, allows interaction via body postures, and facilitates better social expression among users. However, full-body tracking systems today require a complex setup fixed to the environment (e.g., multiple lighthouses/cameras) and a laborious calibration process, which goes against the desire to make VR systems more portable and integrated. We present HybridTrak, which provides accurate, real-time full-body tracking by augmenting inside-out[1] upper-body VR tracking systems with a single external off-the-shelf RGB web camera. HybridTrak uses a full-neural solution to convert and transform users' 2D full-body poses from the webcam to 3D poses leveraging the inside-out upper-body tracking data. We showed HybridTrak is more accurate than RGB or depth-based tracking methods on the MPI-INF-3DHP dataset. We also tested HybridTrak in the popular VRChat app and showed that body postures presented by HybridTrak are more distinguishable and more natural than a solution using an RGBD camera.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; • **Computing methodologies** → *Activity recognition and understanding*.

KEYWORDS

full-body tracking, virtual reality, computer vision.

ACM Reference Format:
Jackie (Junrui) Yang, Tuochao Chen, Fang Qin, Monica S. Lam, and James A. Landay. 2022. HybridTrak: Adding Full-Body Tracking to VR Using an Off-the-Shelf Webcam. In *CHI Conference on Human Factors in Computing*

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04.
https://doi.org/10.1145/3491102.3502045

[1]Inside-out tracking or egocentric pose estimation means that the tracking camera is worn on the user's head. The system locates itself by looking at the environment and locates other body parts according to the camera position. While outside-in tracking means that the cameras are grounded to the environment.

Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3491102.3502045

1 INTRODUCTION

Virtual reality (VR) has great potential in many applications including social networks, gaming, and entertainment. However, current widely adopted VR systems can only track the user's head and hands positions, and not the rest of the user's body. Therefore, most VR apps today either only render the user's upper body or predict the user's lower body position according to their upper body movements. The resulting floating avatars and unsynchronized leg movements may break the user's illusion, hinder the user's expression, and limit the types of apps that developers can build.

However, current full-body tracking solutions have various limitations. Sensor-based or outside-in tracking can produce accurate results, but users need to either wear bulky 3D positional trackers all over their body [10], or use RGBD cameras that require extensive calibration and are hard to acquire [9] (outside-in tracking). On the other hand, recent commercial VR upper-body tracking systems rely on egocentric tracking cameras (referred to as *inside-out tracking*), which require minimal setup, greatly reduce the barrier of entry to VR, and are generally preferred by users[2]. However, research has shown that egocentric cameras cannot provide adequate tracking for the lower body due to intra-body occlusion [1].

Combining the advantages of inside-out and outside-in tracking, we present HybridTrak, which offers an economical, calibration-free, and user-friendly solution for full-body tracking. The novel combination of a single uncalibrated camera and existing inside-out upper-body tracking of HybridTrak is optimal for full-body tracking: the former can see the user's feet without occlusion of the user's upper body (more discussion in Section 5.1); the latter can see the user's hands. HybridTrak first generates 2D full-body poses from the webcam and 3D upper-body poses from an off-the-shelf inside-out tracking system. These data are fed into a pose conversion neural network to produce the lower-body positions and orientations. Combined with the upper body positions from the egocentric cameras, our full-body tracking data can be used by any SteamVR app without requiring any modifications.

We evaluated HybridTrak by objective performance comparison on existing datasets and subjective perception evaluation on pose naturalness and clarity. For objective performance comparison, we found our hybrid tracking setup to be better than using a calibrated RGBD camera with a naïve algorithm for lower-body tracking (on the Human3.6m dataset [14]). We also found our algorithm to be better than a baseline algorithm built with VNect [26] (on the MPI-INF-3DHP [24] dataset). For pose naturalness and clarity, we found that users ($N = 12$) can differentiate five different poses with complex lower-body motion with a higher accuracy using HybridTrak than the other two solutions (RGBD camera and upper-body only tracking). We also found that users rated the poses generated by our system more natural than the baselines.

The contributions of this project include:

(1) A novel system design that can provide a robust and accurate full-body tracking capability for VR with the addition of a single uncalibrated RGB camera.
(2) We introduce a full-neural full-body tracking solution for VR that is more accurate than a baseline that requires an RGBD camera. Whereas the baseline RGBD algorithm gets a Mean Per Joint Position Error (MPJPE) of 0.136m and a Mean Per Joint Rotation Error (MPJRE) of 0.609rad, HybridTrak achieves a better result of 0.098m and 0.282rad, respectively.
(3) A user study using a popular VR chat room application shows that body postures presented by HybridTrak are more distinguishable and more natural compared to an RGBD camera-based tracking system.

2 RELATED WORK

The related work to HybridTrak can be categorized as: 1) Vision-based 3D body pose tracking 2) Non-vision-based 3D body pose tracking, and 3) Other hybrid pose tracking methods.

2.1 Vision-based 3D body pose tracking

Similar to HybridTrak, prior work has tried to use computer vision to detect 3D body poses for a variety of applications. Traditionally, vision-based 3D pose tracking is done with an RGBD camera [9, 19, 43]. However, RGBD cameras usually have a limited range, are error-prone in sunlight, and are not accessible to every VR user. RGB cameras are cheaper and have fewer of those restrictions. Recently, many have researched the area of 2D human body pose estimation using a single RGB camera [3, 5, 8, 40]. However, when it comes to 3D pose estimation with a single RGB camera, it is hard to estimate the size and global position of the skeleton because these systems lack information on the distance between the user and the camera.

Most of the prior work uses either visual cues [2, 25–27], temporal geometry cues [18, 29], or both [6] to deduce a 3D skeleton from one or more 2D images. These algorithms are usually computationally intensive, preventing them from being used in a latency-sensitive scenario such as VR. Also, these algorithms usually predict the body pose in a coordinate system that is relative to one of the joints of the user's body, usually the pelvis, which makes it hard to project the tracking result onto the VR tracking space. Some prior work tries to estimate the global position in real-time by data-fitting the estimated 3D pose with the 2D pose[25, 26], but this requires accurate knowledge of the intrinsic and extrinsic parameters of the camera and is prone to noise in the predicted skeleton sizes, which would result in awkward offsets in the camera direction. In VR, as the user's viewport information is estimated by a much faster and more accurate system (VR tracking), any drift between the full-body tracking and the user's viewport may dislocate the user's body from their head, which is very disturbing for the user. In contrast, HybridTrak processes the image data with the conventional (faster and more accurate) VR upper-body tracking information, which does not require calibration and produces a more accurate and coherent full-body 3D pose estimation.

Besides single-camera 3D pose estimation, researchers have also tried to use multiple cameras for full-body tracking. They usually leverage multiple neural networks to detect 2D poses in each camera

[2]Users show clear preference for inside-out tracking systems as the adoption of these systems increased from 6.4% to 67% in just a year (according to the Steam hardware survey between August 2020 and August 2021 https://store.steampowered.com/hwsurvey)

and fuse the partial results to yield more accurate 3D poses [15, 32]. However, this requires the user to have a calibrated multi-camera setup, which is expensive and hard to configure. Other prior work leverages single or multiple cameras on the user's body [1, 12, 34, 36, 41]. However, due to lens distortion and body obstruction, the accuracy of these systems is still pretty low, especially in the leg area, which is problematic for current VR tracking systems.

2.2 Non-vision-based 3D body pose tracking

Other body pose estimations have been proposed that are not based on computer vision. Commercial motion capture systems, such as Vicon [38] and OptiTrack [28] use multiple cameras and retro-reflective dots positioned on the user's body to accurately track multiple positions on the user's body. While being used as ground truth in many pose tracking datasets, their expensive and complicated setup prevents them from being widely adopted by average VR users. Others have proposed using wearable trackers that are coupled with an external fixed tracking reference, such as solutions presented by Islam et al. [16], Pintaric and Kaufmann [30], and SteamVR tracking [39]. These systems usually offer a limited number of tracking points and require a fixed reference hardware setup and calibration before usage. Other alternative methods have also been proposed, such as using a pressure-sensitive floor [4], radio signals [42], or multiple IMUs [35]. However, the accuracy obtained by these solutions is usually limited.

2.3 Other hybrid pose tracking methods

Researchers have also explored different pose tracking methods by processing input from multiple sources. These approaches are mostly focused on fusing IMU and vision-based pose estimation. Pons-Moll et al. [31] first proposed the idea of combining IMU data with RGB images to produce better tracking results. More recently, researchers proposed using neural networks and information from multiple RGB cameras [11, 23, 37]. Such work improved accuracy by adding inputs in more modalities, but at the cost of more complicated setups. In contrast, HybridTrak improves the accuracy of vision-based tracking by using existing VR tracking systems to minimize the setup cost.

3 SYSTEM DESIGN

We present the system design of HybridTrak in this section with an overview of how the HybridTrak system works, as well as our design rationale, the neural networks used by HybridTrak, and other implementation details.

3.1 Design considerations

HybridTrak aims to provide regular consumers accurate full-body tracking with minimal setup overhead. HybridTrak uses a single uncalibrated webcam and a common inside-out upper body tracking system. The user only has to place a camera in a place where the user's body can be seen without occlusion, put on the VR headset and controllers, and enter the VR as usual. To give the user a seamless and responsive experience with HybridTrak, we designed the system with the following goals:

(1) Calibration free (no need for the extrinsic and intrinsic matrix of the camera),

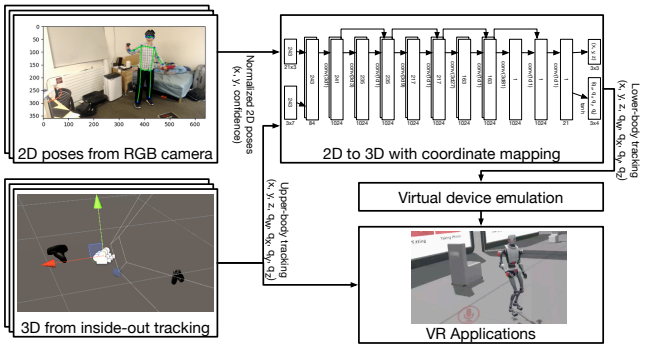

Figure 2: System architecture of HybridTrak: HybridTrak first processes the webcam footage to extract normalized 2D poses. HybridTrak's pose-conversion neural network then processes the 2D poses from the webcam and the upper-body tracking coordinates from an existing VR tracking system and converts the 2D poses to lower-body 3D poses in VR coordinates. Finally, HybridTrak can emulate virtual tracking devices with the user's lower body tracking points and pass through the data from inside-out tracking devices for the upper body. In this way, VR applications that support full-body tracking on SteamVR can work without modification.

(2) Accurate global positions, and
(3) Provide input that is compatible with existing VR systems.

To render the body pose in VR properly, we need the user's pose in global coordinates in the real world space (z-axis is up). Also, we wish to minimize the calibration process. While most prior work can only predict the body pose *relative* to a body joint with an RGB camera, some prior work [24] demonstrated methods to get the global coordinates from 2D images. However, their method relies on the accurate intrinsic and extrinsic matrix of the camera as well as an estimation of the body skeleton of the user from the 2D image, which is not always reliable. The resulting 3D poses may have accuracy and latency that is acceptable for 2D games like those built for the Kinect, but they do not meet the standard of keeping people immersed and preventing them from getting dizzy in VR. If these poses are used directly with existing VR headset tracking, without calibration and without reliable skeleton size estimation, the projected pose positions in world coordinates are likely to have an offset from the user's headset position, which is very disorienting. Since inside-out/egocentric tracking for the upper-body is common and effective in commercial systems, we leverage the upper-body tracking data to provide a calibration-free and accurate lower-body pose estimation. In upper-body tracking systems, the controllers held by the users contain markers and a motion sensor. thus they can track the hand positions more accurately and reliably than image-based pose estimation. We use those tracking points to project the detected 3D pose back to the VR space.

HybridTrak is designed to work with existing VR systems to provide the full-body 3D poses to VR applications. Currently, most of the VR applications with full-body support use what is called a *six-point format*[3]. It consists of the position and orientation of the

[3] see https://docs.vrchat.com/docs/full-body-tracking

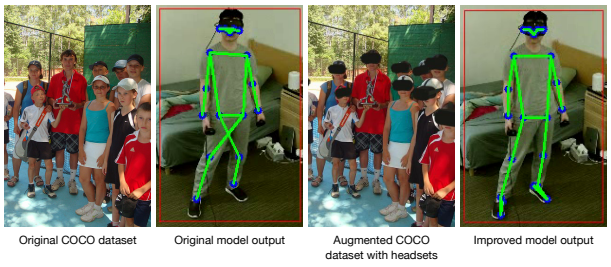

Figure 3: We augmented the COCO dataset with images of people wearing VR headsets and re-trained an improved pose_resnet model with four more key points in the output. The augmented dataset solves the problem that the original model may treat part of the user's body as flipped when the user's face is blocked by the VR headset. The added key points allow HybridTrak to accurately estimate feet orientation.

user's head, waist, two hands, and two feet. However, most pose-tracking algorithms today predict only the position, and not the orientation, of many more joints in the user's body, while existing VR games want the position and orientation of a smaller number of joints. HybridTrak is fully compatible with existing VR applications as it directly generates the waist and feet tracking points that include both the position and the orientation. HybridTrak can also be configured to output more position points to provide a more accurate estimation of the user's full skeleton.

3.2 System architecture

With these design considerations, the overall architecture of HybridTrak is shown in Figure 2. HybridTrak accepts input from two sources: 2D poses from an RGB camera and 3D upper-body inside-out tracking data. To generate the 2D poses input, we use a modified version of pose_resnet [40] that works well with users' images (even when the VR headset is blocking their face) and can output extra key points to provide the necessary foot orientation for VR full-body tracking. For 3D poses, we use the internal headset and controller tracking provided by the Oculus Quest. We use a pose-conversion neural network that accepts the 2D poses and the 3D upper body VR tracking data to produce 3D poses. By leveraging the temporal information from 2D poses and 3D upper-body tracking data, HybridTrak can generate accurate 3D poses in VR space without requiring prior calibration. Finally, we present the generated 3D poses as virtual trackers to SteamVR, which allows unmodified applications to read lower body tracking data from virtual trackers generated by HybridTrak and get upper-body tracking data from existing headsets and controllers.

3.3 HybridTrak algorithm

The HybridTrak algorithm consists of two steps:

(1) Generate 2D poses from the webcam.
(2) Map the coordinates of the 2D poses to those of the 3D poses.

3.3.1 Generating 2D Poses. Common 2D pose detectors usually output 2D poses in the 17-key point COCO format [17, 22]. Although these points are sufficient for representing positions of the user's body parts, they lack information about the orientation of the user's limbs. For HybridTrak, we especially care about the user's feet orientation, as they represent 6 out of the 18 degrees of freedom in the user's lower body. Another problem regular pose detectors have is that their accuracy sometimes relies on the fact that the user's face is uncovered. In practice, we observed that a vanilla pose estimation model tends to predict that the user is facing backwards when the VR headset is blocking their face.

To address this problem, we trained our improved version of the pose_resnet model [40]. Pose_resnet is a 2D pose estimation neural network that uses a ResNet as the backbone for feature extractions and adds a few deconvolutional layers over the last convolution stage in the ResNet. It can provide great accuracy even when the user has overlapping body parts and is standing in front of a complex background. We added to pose_resnet extra feet key points from the COCO-Wholebody [17] dataset. We also augment the training data by generating images with an overlaid headset from the original COCO dataset. To do so, we fitted a VR headset onto a human head model and measured the 3D positions of the key points on the head (nose, eyes, and ears from the COCO key points, eye corners, ear corners, and chin from COCO-Wholebody key points). We computed the 3D pose estimation for the headset from these key points and overlaid the projected headset model back onto the images in the COCO dataset. The results of the model before and after our modification are shown in Figure 3.

On a side note, pose_resnet is chosen because it strikes a good balance between accuracy and speed. For comparison, we tried other top-down 2D body pose trackers like HRNet or OpenPose for 2D pose tracking; however, they are noticeably slower than pose_resnet and offer limited accuracy improvement for our use cases (average precision from 73.7 to 77.0 on COCO test set for HRNet). Notably, HRNet is less computationally intensive in theory, but it runs slower on current hardware (acknowledged by the authors on GitHub[4]). Other bottom-up pose trackers perform better when there are multiple people, but that is rarely the case for VR pose tracking. In our early experiments, we also found top-down pose trackers to be more robust against self-occlusion, which is common for VR.

We evaluated the accuracy of the model based on accuracy on the COCO evaluation dataset with our headset augmentation. We found that the model trained on our augmented dataset yields an average precision (see COCO human pose benchmark [22] for definition) of 72.2, while a baseline model trained on the original COCO dataset had an average precision of 65.4.

3.3.2 2D pose to 3D pose with coordinate mapping. The key challenge in HybridTrak is the mapping of the 2D pose coordinates to lower-body 3D poses that are consistent with the 3D upper-body tracking points (VR coordinates). We train a pose conversion neural network to directly process 2D poses from the webcam and 3D tracking data from the inside-out upper-body tracking system and use those to output the 3D poses in VR coordinates.

[4] see https://github.com/leoxiaobin/deep-high-resolution-net.pytorch/issues/26

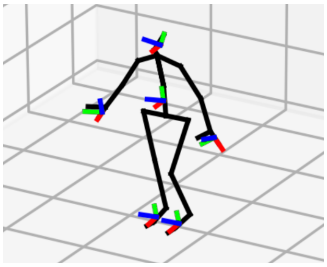

Figure 4: Converting VR tracking points from the Human3.6m dataset. The black skeleton represents the Human3.6m dataset, and the RGB axes represent VR tracking points. Red is front, blue is right, and green is up.

We used an existing 3D pose estimation dataset Human3.6m to train this model. Although this dataset does not have tracking data in the VR coordinate space, we compute the position and the orientation of the tracking points used in VR from the position of the existing annotated key points. For VR, the tracking points include the head, two hands, waist, and two feet. For the head, we compute the position by projecting the `face`[5] point to the line segment between `head_top` and `neck`. For hands, we compute the positions by getting the middle points between `index_finger` and `wrist`. For waist, we computed the positions by getting the weighted average point between `neck_center` (weight 1) and `hip_center` (weight 3). For feet, we compute the midpoint between `ankle` and `toe`. Finally, we use joints close to these points to compute the orientation of those points. The final result is shown in Figure 4. The generated head and hand points are used as input to the network and the generated waist and feet points are used as reference output to the network.

To train the network, we also generate the 2D pose from videos captured in Human3.6m with the modified pose_resnet estimator with feet key points (as described in Section 3.3.1). To ensure that our model can generalize to an arbitrary camera configuration that a user may have, we also normalize the 2D pose points to make the result in the longer axis in x-y between $[0, 1]$, and keep the aspect ratio the same while scaling the other axis. In this way, even if the user's camera has a different focal point or pixel density (camera intrinsic matrix), it should not affect the scale of the 2D pose.

We also apply a random rotation along the z-axis and a random offset on the xy-plane on the generated data as a data augmentation method to make sure that the model can handle arbitrary offsets and rotations between the camera's viewpoint (extrinsic matrix) and VR coordinates. This allows our model to work out of the box with no prior calibration.

We adopt an architecture that is similar to VideoPose3D [29], as shown in Figure 5. The network accepts 2D poses p from our modified pose_resnet, along with three of the upper-body inside-out tracking points (head and hands) u, and produces three outputs (feet and waist) l with position and orientation. It uses a fully convolutional architecture with residual connections. In training, we compute 2D poses p from images using the same modified pose_resnet while calculating the upper-body u and the lower-body l tracking points from tracking points of the Human3.6m dataset. We use the 3D u and 2D tracking points p as input for the pose conversion neural network. We then compare the output VR tracking points l' with the generated lower-body tracking points l with a Mean Per Joint Position Error (MPJPE) as loss for positions and a Mean Per Joint Rotation Error (MPJRE) as loss for orientations.

3.3.3 Implementation. For pose_resnet, we use YOLOv3 [33] as the person detector for detecting the bounding box for pose_resnet, and we use a modified 384x384 Resnet152 variant of the pose_resnet model trained on our augmented COCO dataset with the COCO-WholeBody annotations. We trained the HybridTrak's pose conversion neural network with 160 epochs, learning rate at 0.001, and decay of 0.95 on every epoch. The training takes around 10 hours on a machine with NVIDIA Tesla V100. For VR input and device emulation, we communicate with OpenVR through its API. We emulate three virtual trackers from our three predicted tracking points, which make this system compatible with almost all VR programs supporting full-body tracking on the SteamVR store. We tested the system on a computer equipped with an Intel i7-8700k processor and an Nvidia RTX 2080Ti graphics card and we observe that we can stably process the camera image from a Logitech C930e webcam in 30fps. The frame processing latency of HybridTrak from RGB image to 3D pose in VR averages 0.0827s, and the jitter is 0.0063s.

4 EVALUATION

We evaluate the effectiveness of HybridTrak in two ways: 1) objective performance comparison based on an existing dataset and 2) subjective perception of pose naturalness and clarity by users in a VR social network.

4.1 Comparison with RGBD-camera-based algorithms

In this section, we discuss the performance comparision with RGBD-camera-based algorithms. To evaluate the overall performance of our system, we test the accuracy of the system in predicting the waist and feet positions and orientations in the Human3.6m data set. We use P9 and P11 in Human3.6m as the test set and the other nine participants as a training set for HybridTrak.

RGBD-camera-based solutions are common among VR users. These solutions also use cameras on the headset to track their upper-body poses while using *calibrated* external RGBD camera to track their full-body poses. However, unlike HybridTrak, these systems ignore the upper body tracking points from the RGBD cameras and only use the lower body tracking points with a fixed transformation provided by an extra calibration step.

We set up a *Virtual RGBD* baseline by aligning the time-of-flight camera image in Human3.6m with the RGB camera image to form a virtual RGBD camera, and use the body skeleton detected by the same modified pose_resnet model (as described in Section 3.3.1) to generate 2D poses. Using a naive lifting method, similar to prior work [43], we can extract 3D poses from the depth image. Then we compute the positions and orientations for the lower-body VR tracking points as described in Figure 3.3.2.

[5] All the "`equal width font`" words denote joints in the Human3.6m dataset.

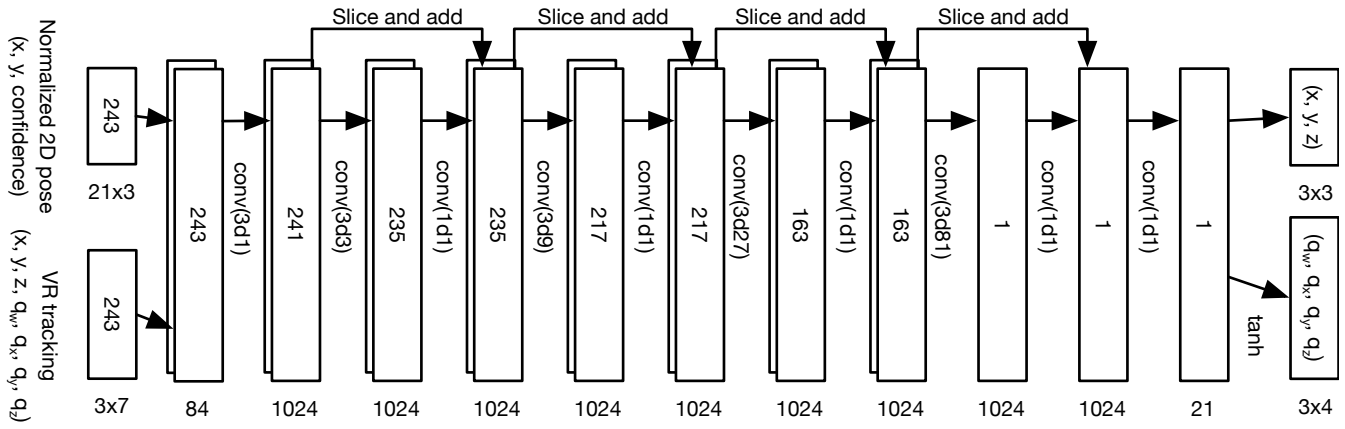

Figure 5: HybridTrak's pose conversion neural network: The model accepts VR tracking points for the head and hands and 2D pose estimation as input, processes it using a temporal CNN, and outputs full-body tracking results for the waist and feet. conv(*n*d*m*): 1D convolution layer with a kernel size of *n* and a dilation of *m*; Slice and add: we slice the output of one layer symmetrically and add the residue to the output of another layer; Boxes represent the feature vectors (numbers in the box are the feature size; numbers under the box are the number of channels.); All unmarked arrows are fully-connected linear layers. We use ReLu for activation across all layers. Batch normalizations are added between convolutional layers.

We also created an alternative method *HybridTrak-transform* (see the detailed implementation in Appendix A) to test against the full-neural solution of HybridTrak. In this method, we first convert 2D poses to 3D poses in camera space using a neural network that is similar to VideoPose3D [29]. We then use "least-squares fitting" to match the head and hands tracking points to the data from inside-out tracking and produce a transformation between the camera space and the real-world space. Finally, we apply that transformation to the predicted waist and feet poses to produce data for VR full-body tracking.

The results are shown in Table 1. HybridTrak shows the best result for both position and orientation, and HybridTrak-transform is second best, while the Virtual RGBD baseline gives the worst result. The RGBD baseline results are comparable to prior work [43]. The rotation error is especially large for the RGBD baseline. Having the feet and the waist pointing in the wrong direction can be very disturbing for the VR user. We think that the large rotation error is due to two reasons: 1) Both HybridTrak algorithms have a temporal CNN to correct for temporal inconsistency, while the RGBD baseline does not. 2) RGBD cameras may give a wrong estimation of the position of the body parts that have been occluded. Note that due to the labeling differences in the training data of pose_resnet and our test set from Human3.6m, there may be a small systemic offset between pose_resnet annotation and the ground truth, causing a higher error in the evaluation result. Nevertheless, both HybridTrak methods perform well given that all the coordinates are in global coordinates and our HybridTrak methods do not need to know the ground-truth transform between the camera space and the real-world space.

When comparing within the two implementation of HybridTrak, we noticed that HybridTrak's full-neural solution is more robust to pose tracking noise and requires less compute power, so we used the full-neural solution in our user study. On the other hand, HybridTrak-transform computes the transform matrix between the camera space and the world space, which is useful on its own. We can reverse the transform matrix to project the 3D positions of the user's head and hands to the camera space and reduce the workload or improve the result of person bounding-box detection, which is a crucial step in 2D pose detection. Since an RGB camera can also be used to track objects of given sizes, with the help of this deduced camera to VR transform, we can use it to project other objects in the real world back to VR. These applications are harder to achieve with a full-neural solution.

One limitation of the Human3.6m dataset is all the camera positions are at the same height (camera pitch). To test the performance of our system with a wider range of camera pitch, we also trained and tested the HybridTrak model on the MPI-INF-3DHP [24] dataset, which has more variety of camera heights. One thing notable is that in the MPI-INF-3DHP dataset, the user is sometimes out of the camera frame, in that case, we still feed the 2D key point detection results with low confidence scores to the HybridTrak model. Our model shows a comparable result on the MPI-INF-3DHP dataset with the result on the Human3.6m dataset despite the constraints of the dataset, which means that the model can perform relatively well when the camera pitch changes.

4.2 Comparison with other RGB-camera-based algorithms

One major feature of HybridTrak is that we aggregate the upper-body inside-out 3D tracking data with the full-body webcam 2D tracking data over a period of time and fully integrate them in the system's neural network. To evaluate whether this system design is beneficial, we compared our system with a baseline that directly produces 3D relative tracking points from the RGB camera,

Table 1: Overall performance comparison: We compared HybridTrak and HybridTrak-transform with a baseline based on data from a virtual RGBD camera. Both of our algorithms performed better than the baseline. Among our two algorithms, HybridTrak has a slightly better result on both position and orientation prediction. The best result on this metric is underlined.

Model name	Dataset	GT transform	Position Error MPJPE(m)	Rotation Error MPJRE(rad)
Virtual RGBD	Human3.6m	YES	0.136	0.609
HybridTrak-transform	Human3.6m	NO	0.104	0.306
HybridTrak	Human3.6m	NO	<u>0.098</u>	<u>0.282</u>
HybridTrak	MPI-INF-3DHP	NO	0.123	0.350

Table 2: Global coordinates performance comparison: We compared the results of a variant of the HybridTrak algorithm that predicts 17-joint global positions similar to prior work. We found that our algorithm performed better in terms of position error on the MPI-INF-3DHP dataset compared with VNect [26].

Model name	Dataset	Position Error MPJPE(m)
VNect	MPI-INF-3DHP	0.455
HybridTrak (17 joints)	MPI-INF-3DHP	<u>0.138</u>

and then simply uses head position to compute the absolute positions of these tracking points. The baseline solution is built with VNect [26] on MPI-INF-3DHP. Although the original VNect papers presented methods of estimating the global positions of joints, we found that although the output global coordinates are consistent with themselves over time, they are not always aligned with the ground truth. So the baseline solution first estimates the relative 3D positions with VNect, then computes the global coordinates by aligning relative positions with the ground-truth head positions. For HybridTrak, we retrained our model to accept 2D positions from the camera footage in the MPI dataset as well as the 3D head and hands tracking data to produce all of the 14-joint positions in MPI-INF-3DHP. The head and hands tracking points we provided to HybridTrak are `head` (7th), `left_hand` (13th), `right_hand` (18th) points in the MPI-INF-3DHP, respectively.

The results are shown in Table 2. HybridTrak has much better performance in terms of position error than the baseline system based on VNect. Note that the VNect paper reported an MPJPE of 142mm, but the result listed in their paper is computed using the ground-truth bounding boxes and waist (`pelvis`) 3D positions. Since this information is not available in real-world live-inferencing, we used the same bounding boxes generated by the YOLOv3 for 3D pose estimation in VNect (the same bounding boxes for 2D pose estimation in HybridTrak). For global position estimation, we used head position to convert relative coordinates to global coordinates. Although not available in real-world usage, we also conducted the evaluation when ground-truth pelvis position is used for global position estimation. The MPJPE in this case is 0.325m; it is slightly better than using head positions, but still worse than HybridTrak.

These results show that the architecture of HybridTrak can effectively leverage the power of inside-out upper-body tracking and data from the external camera to generate more accurate global positions for full-body tracking in VR.

4.3 User Study

One important use case of full-body tracking is to provide better social interaction between users in VR. We conducted a user study in VRChat[6], a VR social network that supports full-body tracking, to see if our 3D pose in VR is natural and distinguishable from another chat user's perspective. VRChat, like many other programs supporting full-body tracking on SteamVR, expects three additional tracking points with position and orientation from the SteamVR driver, one for the waist and two for the legs. HybridTrak can emulate these tracking points using a custom-made SteamVR driver (as described in Figure 2).

We compared HybridTrak both with another popular trackerless full-body tracking system called KinectToVR[7] and with upper-body only tracking, which is similar to most commercial VR products. In the upper-body only tracking condition, we do not feed any lower body tracking points to SteamVR, and VRChat has an internal mechanism to generate a lower body posture that fits the position and orientation constraints of the user's head and hands.

4.3.1 Task. The experimenter invites the participants into a special VRChat room built for this study, After the consent forms are filled out, the experimenter gathers basic demographic information from the participants. We then ask the participant to evaluate three full-body tracking systems: 1) HybridTrak, 2) KinectToVR, and 3) upper-body only tracking. The three systems are presented to users in a counter-balanced order. In each case, the experimenter performs 15 poses (five different poses, each pose performed three times) in random order while using the tracking system currently being evaluated. We selected five representative poses from an existing 3D pose tracking dataset [14, 24] with distinct leg postures and similar upper-body positions (see Figure 6), so as to highlight differences in lower-body tracking.

In VR, the participants can see an image with the five reference poses (top row in Figure 6). After each pose is presented using the experimenter's VR avatar, the experimenter asks the participant to identify which pose it is. Usually, camera-based pose tracking methods show a better result at the same angle as the capture camera. We asked the participants to observe the experimenter at any position that they felt comfortable in. This allows participants to evaluate the pose generated by HybridTrak at an arbitrary angle in 3D space instead of viewing it from the same angle as the webcam (in HybridTrak)/Kinect being used for tracking to demonstrate the true performance of the tracking system in VR.

[6] https://www.vrchat.com
[7] KinectToVR Kinect Full-Body Tracking: https://k2vr.tech

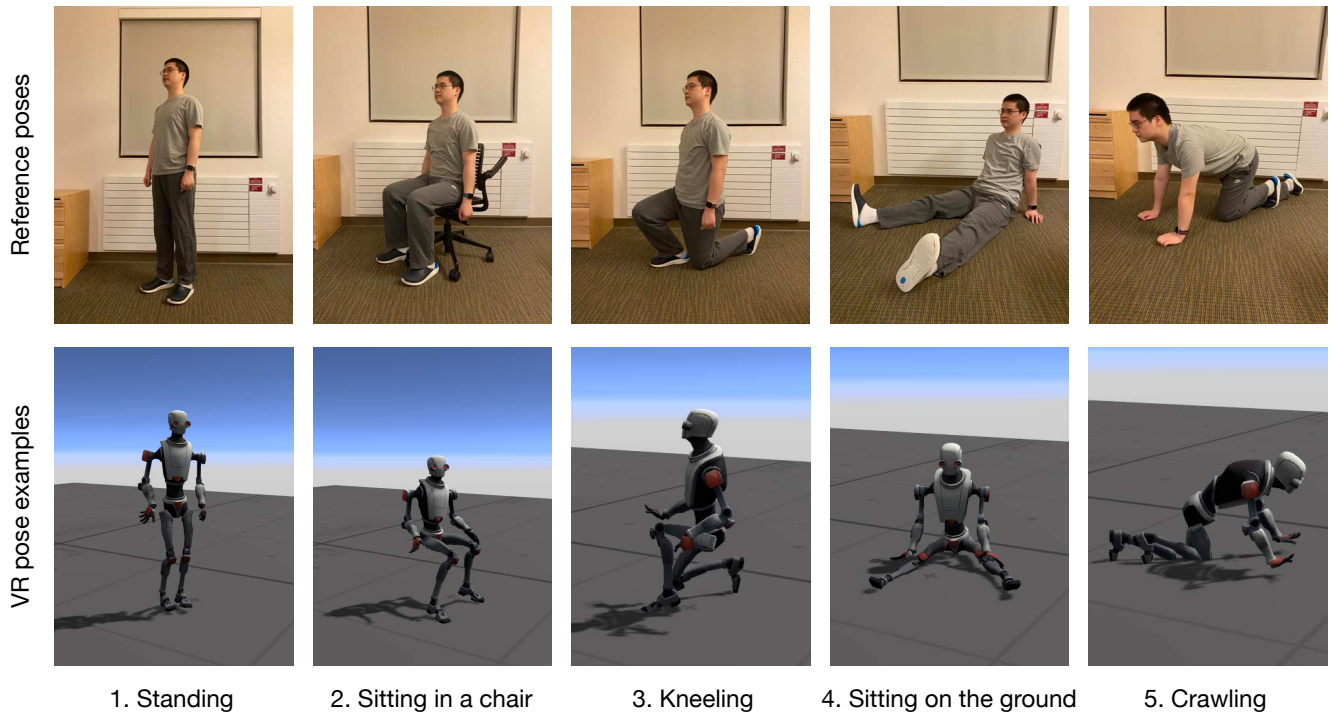

Figure 6: Poses used in the user study: The experimenter performed 15 poses in random order (each of the five poses is presented three times) and asked participants which pose they thought the experimenter was presenting. The user can view the experimenter's posture at any angle that they feel comfortable with. In the end, we asked them about their overall perception of the presented full-body tracking systems.

After the participants see how each system performs, the experimenter asks the user about whether they agree or disagree with the following statements (7-point Likert scale, from "Strongly disagree" to "Strongly agree"): 1) "The presented body postures are natural." 2) "The transitions between body postures are natural." At the end of each condition, we also ask the participants for subjective feedback on the current system. At the end of the user study, we ask the participants which system they would rate the best and why.

4.3.2 Participants. We recruited 12 participants (six female) in our study, aged between 18-52 (median 25.5). Most of our participants were frequent VR users, with four of them using VR weekly, three of them using VR daily, and two of them using VR monthly. The other three participants only use VR a few times a year. Most users have little experience using full-body tracking systems. Eight of our participants have never used full-body tracking systems, two use these systems a few times a year, and two use full-body tracking systems monthly.

4.3.3 Results. The results of all the pose identification responses across users are shown in Figure 7. Participants identified 99% of the poses correctly in the HybridTrak condition, while there were more misidentifications made in the KinectToVR and no full-body tracking conditions. We also computed the recognition accuracy for each user in each of the three systems (Figure 8) and ran paired t-tests between HybridTrak and the other two baseline systems. We

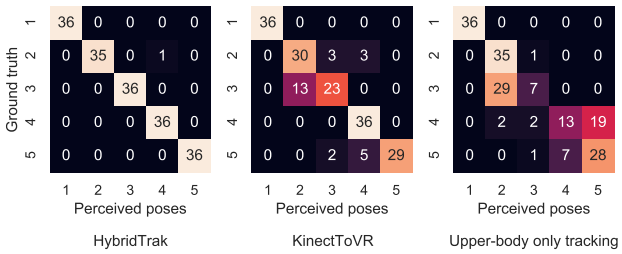

Figure 7: Confusion matrix of the perceived experimenter poses for the three tracking solutions. 1-5 corresponding to the poses in Figure 6. Only one out of 180 responses for the HybridTrak solution was incorrect.

found that participants can identify poses significantly better in the HybridTrak condition than in the KinectToVR condition ($t = 3.84$, $p = 0.0028$) and in the upper-body only tracking condition ($t = 9.31$, $p = 1.5e-6$).

The results of the user's perceived naturalness of the poses and the naturalness of the transitions between poses is shown in Figure 9. We computed paired t-tests and found statistically significant differences between HybridTrak and the baseline conditions in terms of posture naturalness ($t = 8.62$, $p = 3.2e-6$

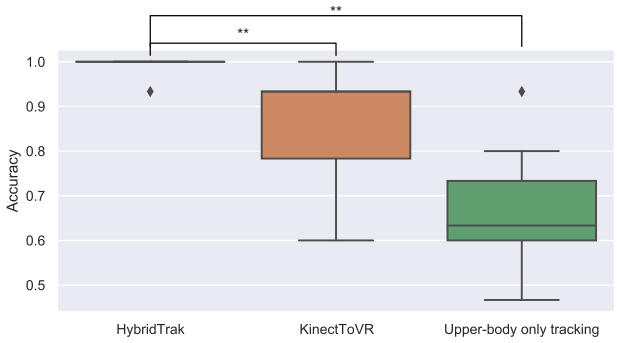

Figure 8: Accuracy of perceived poses for the three tracking solutions: We found the pose perception accuracy in the HybridTrak condition is significantly higher than in the KinectToVR and in the upper-body only tracking systems. **: p < 0.005

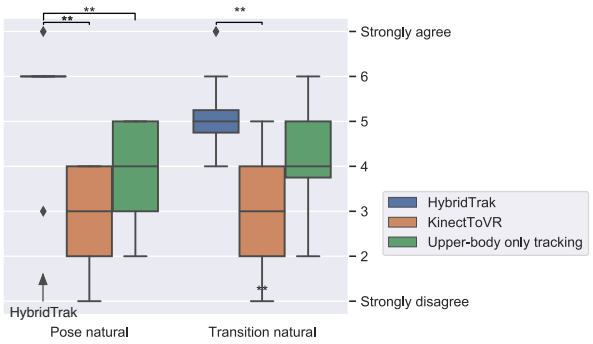

Figure 9: Naturalness of the presented postures and transitions between them for the three tracking solutions: Participants rated poses presented by HybridTrak to be more natural than the other solutions. They also found the transitions in the HybridTrak system to be more natural than those in the KinectToVR system. **: p < 0.005

and $t = 6.14$, $p = 7.3e-5$). We also found a statistically significant difference between HybridTrak and KinectToVR in terms of transition naturalness ($t = 6.19$, $p = 6.7e-5$), but the difference between HybridTrak and upper-body only tracking was not significant ($t = 1.89$, $p = 0.085$). We think the reason for the small differences in perceived transition naturalness between HybridTrak and upper-body only tracking is that when VRChat does not have lower body info, it generates smooth-looking transitions even if they do not represent the actual state of the experimenter's lower body movements.

We also collected subjective feedback from the participants. All twelve participants rated HybridTrak to be the best of the three systems. P1, P2, P8, P9, and P11 found the poses in the HybridTrak condition to be clear and natural. P2, P10, and P11 found the transitions between poses in this condition to be sometimes jerky, especially when transitioning between crawling and sitting on the ground (P2). For the KinectToVR condition, P5, P6, P7, P10, and P11 found the leg orientations in the KinectToVR condition to be inaccurate, which is a common complaint about Kinect body tracking. For the upper-body only tracking condition, P4-9 and P12 found the postures to be less distinguishable. P5 reported that they had to guess the lower body posture from the upper body posture.

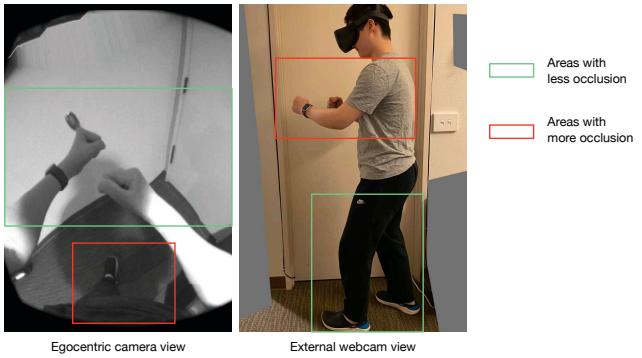

Figure 10: Captured images from an egocentric camera and an external webcam. It shows that the user's lower body is likely to be occluded with an egocentric camera and not with the external webcam, while the opposite is true for the user's upper-body.

5 DISCUSSION

In this section, we discuss how HybridTrak solves the occlusion problem, the comparison between HybridTrak and other 3D pose tracking algorithms, applications of our algorithm, avenues for future work, and the limitations of HybridTrak.

5.1 Occlusions and benefits of a hybrid tracking setup

Occlusions are a common source of errors for most pose-estimation algorithms [7]. HybridTrak is designed to avoid occlusions. There are two common types of occlusion in pose estimation: occlusion between body parts and occlusion due to external objects. The latter is not an issue for VR, since VR requires the tracking space to be clear of objects to keep the user safe.

For occlusion between body parts, the hybrid tracking architecture of HybridTrak cleverly avoids most of the problems with a minimal setup overhead. As shown in Figure 10, the user's lower body is likely occluded in the egocentric camera (including multiple fisheye cameras) and not occluded in the external webcam, while the opposite is true for the user's upper body. By combining egocentric upper-body pose detection with a lower-body pose from an external camera, HybridTrak achieves full-body tracking with a simple and portable setup.

5.2 Comparison between HybridTrak and other 3D pose tracking algorithms

We have demonstrated that: 1) HybridTrak can produce more accurate 3D poses than a naïve algorithm using RGBD input (Section 4.1); and 2) HybridTrak's generated poses are perceived by users to be

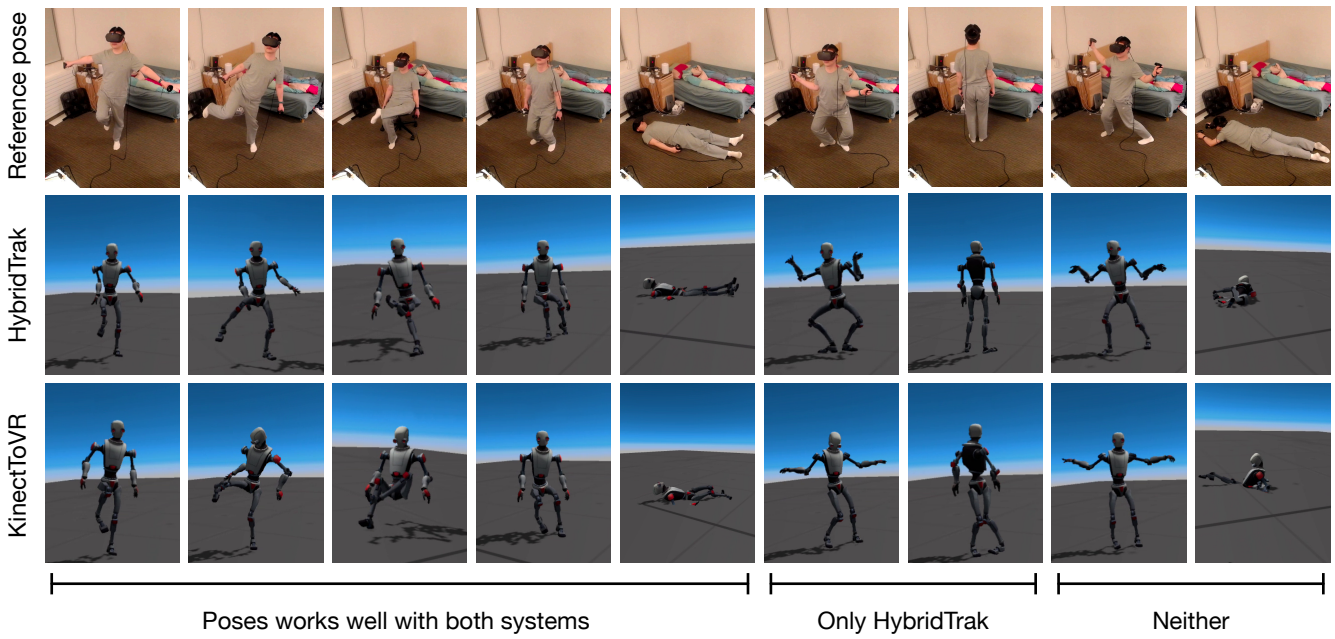

Figure 11: Generated pose comparison between HybridTrak and KinectToVR. Some poses work well in both systems, some work only in HybridTrak, and some work in neither system. The three rows are not captured at the same time, so there may be some differences in the exact position of the arms and legs. In the 7th pose, the user is facing backward while the legs in the KinectToVR solution are facing forward. In the 8th pose, the user's right leg is bending forward while his left leg is bending towards the right.

more accurate and more natural than those from an RGBD camera-based solution, KinectToVR (Section 4.3). Since many VR users are already using KinectToVR for body tracking, this demonstrates that HybridTrak can reach a level of performance that is beneficial to many current VR users using only a single webcam. However, this does not mean HybridTrak is incompatible with the RGBD camera. Future work can integrate the depth info from such a camera into the detected 2D poses and feed the combined information into a similar pose conversion neural network as that used by HybridTrak to potentially achieve even more accurate 3D poses in VR.

Specifically, when comparing with KinectToVR, we found that HybridTrak can accurately reproduce more poses than KinectToVR, especially when the user's feet are facing sideways (Figure 11). We think this is because the Kinect is mostly using depth to extract the user's skeleton, but feet provide little contrast in terms of depth. In contrast, HybridTrak uses RGB information to extract the user's feet positions and direction, which is more reliable for determining the position of the feet.

As shown in Figure 11 even HybridTrak does not work well with some poses. However, as HybridTrak only needs 2D key points as input and 3D key points as output for training data, it is possible to build a synthetic dataset with these postures to enhance the accuracy of HybridTrak. Furthermore, as modern VR apps already contain many character animations, we imagine future VR apps can ship with specialized HybridTrak models. These models can be trained with the included application-specific animations, so that HybridTrak can accurately produce the postures that are common in these apps. This is a benefit specific to HybridTrak's current architecture. Compared to an alternative system that directly generates 3D poses from RGB images, we can harvest the larger annotated image datasets with 2D poses to accommodate different lighting conditions of the users and synthesize arbitrary 3D pose-only datasets to improve the recognition results for specific poses.

5.3 Applications

HybridTrak offers an easy-to-access solution for everyday users to achieve full-body tracking. With HybridTrak, users can have a better experience in social apps now that the full-body posture is accurately presented. It can also provide full immersion for sports such as soccer or dodge ball. HybridTrak can even be used to facilitate the fundamental interactions in VR, such as locomotion. Some locomotion methods, such as Seven League Boots [13], can provide users a better experience when the locomotion method can leverage accurate foot movements.

5.4 Future work

Like any other system using machine learning, HybridTrak would benefit from more training data. A unique benefit of HybridTrak is that the model's input and output can be easily collected from an RGB camera and a commercial full-body tracking system such as HTC Vive Trackers [10]. So an interesting future project would be to crowdsource training data from people who already have a

webcam and a full-body tracking system, which is popular in online VR platforms (6 out of 8 of the participants in our user study have access to a full-body tracking system regularly).

5.5 Limitations

HybridTrak in its current state requires one dedicated graphics card for the tracking system. Note that an accurate 2D pose estimator, such as pose_resnet, requires most of the computing resources in the entire HybridTrak system; the pose conversion neural network we introduce in HybridTrak only requires minimal resources. We tested a variant of HybridTrak that runs the 2D pose detection on an iPhone and streams the result back to a VR-capable machine. This system runs the pose conversion network on the VR machine. While the pose conversion network can run at 30fps, the 2D pose detector is limited to 11fps on an iPhone 11. With a mobile-optimized 2D pose detector and more capable hardware, future all-in-one VR headsets like Oculus Quest would be able to achieve smooth full-body tracking with an extra smartphone on a stand running HybridTrak.

We compared the performance of our system with VNect [26] in Section 4.2. We used VNect as a baseline because it is comparable with HybridTrak in terms of computing resources required and inference latency. Most other pose estimation models either cannot estimate 3D poses with a single camera (e.g., OpenPose [5]), or demonstrate live-inferencing capability (e.g., SPIN [21]). Notably, VIBE [20] can run at 30fps on a modern graphics card and has better MPJPE but worse PCK and AUC scores than VNect. We did not compare with VIBE in this paper, but it could be another alternative RGB pose tracking algorithm similar to VNect.

The pose conversion neural network in HybridTrak is trained with the Human3.6m dataset, which has a limited set of body skeleton sizes. A person with a very large or small skeleton may experience a higher error rate than other people. A possible solution is to apply a random scaling factor to the body skeleton in the training data, and scale the 3D ground truth and the detected 2D pose from the RGB camera accordingly.

6 CONCLUSION

HybridTrak shows a promising future where every VR user can have their full-body represented in the virtual world, by adding just a single off-the-shelf webcam. By making this technology more accurate and versatile, future VR systems will be able to provide a more immersive and interactive environment.

ACKNOWLEDGMENTS

The authors would like to acknowledge Tianshi Li for her help with the early prototypes and the writing of this paper. Also, the authors would like to thank the help of Zennon Melnyk and Daniar Imanbayev for giving early feedback. Finally, the authors would also like to thank the reviewers for their constructive feedback.

This work is supported in part by the National Science Foundation under Grant No. 1900638.

REFERENCES

[1] Karan Ahuja, Chris Harrison, Mayank Goel, and Robert Xiao. 2019. MeCap: Whole-Body Digitization for Low-Cost VR/AR Headsets. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM. https://doi.org/10.1145/3332165.3347889

[2] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. 2016. Keep It SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. *Lecture Notes in Computer Science* (2016), 561–578. https://doi.org/10.1007/978-3-319-46454-1_34

[3] Lubomir Bourdev and Jitendra Malik. 2009. Poselets: Body part detectors trained using 3D human pose annotations. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE. https://doi.org/10.1109/iccv.2009.5459303

[4] Alan Bränzel, Christian Holz, Daniel Hoffmann, Dominik Schmidt, Marius Knaust, Patrick Lühne, René Meusel, Stephan Richter, and Patrick Baudisch. 2013. GravitySpace: tracking users and their poses in a smart room using a pressure-sensing floor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. ACM Press. https://doi.org/10.1145/2470654.2470757

[5] Zhe Cao, Gines Hidalgo Martinez, Tomas Simon, Shih-En Wei, and Yaser A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), 1–1. https://doi.org/10.1109/tpami.2019.2929257

[6] Yu Cheng, Bo Yang, Bo Wang, and Robby T. Tan. 2020. 3D Human Pose Estimation Using Spatio-Temporal Networks with Explicit Occlusion Training. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 07 (Apr 2020), 10631–10638. https://doi.org/10.1609/aaai.v34i07.6689

[7] Yu Cheng, Bo Yang, Bo Wang, Yan Wending, and Robby Tan. 2019. Occlusion-Aware Networks for 3D Human Pose Estimation in Video. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. https://doi.org/10.1109/iccv.2019.00081

[8] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. 2005. Pictorial Structures for Object Recognition. *International Journal of Computer Vision* 61, 1 (jan 2005), 55–79. https://doi.org/10.1023/b:visi.0000042934.15159.49

[9] Full-Body Tracking for Kinect | KinectToVR EX [n.d.]. https://k2vr.tech/. (Accessed on 09/17/2020).

[10] HTC. [n.d.]. VIVE™ | VIVE Tracker. https://www.vive.com/us/accessory/vive-tracker/. (Accessed on 09/17/2020).

[11] Fuyang Huang, Ailing Zeng, Minhao Liu, Qiuxia Lai, and Qiang Xu. 2020. DeepFuse: An IMU-Aware Network for Real-Time 3D Human Pose Estimation from Multi-View Image. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. https://doi.org/10.1109/wacv45572.2020.9093526

[12] Dong-Hyun Hwang, Kohei Aso, and Hideki Koike. 2019. MonoEye: Monocular Fisheye Camera-based 3D Human Pose Estimation. *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (Mar 2019). https://doi.org/10.1109/vr.2019.8798267

[13] Victoria Interrante, Brian Ries, and Lee Anderson. 2007. Seven League Boots: A New Metaphor for Augmented Locomotion through Moderately Large Scale Immersive Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces*. IEEE. https://doi.org/10.1109/3dui.2007.340791

[14] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 7 (Jul 2014), 1325–1339. https://doi.org/10.1109/tpami.2013.248

[15] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. 2019. Learnable Triangulation of Human Pose. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. https://doi.org/10.1109/iccv.2019.00781

[16] Shahidul Islam, Bogdan Ionescu, Cristian Gadea, and Dan Ionescu. 2016. Full-body tracking using a sensor array system and laser-based sweeps. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. https://doi.org/10.1109/3dui.2016.7460034

[17] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. 2020. Whole-Body Human Pose Estimation in the Wild. *Lecture Notes in Computer Science* (2020), 196–214. https://doi.org/10.1007/978-3-030-58545-7_12

[18] Isinsu Katircioglu, Bugra Tekin, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. 2018. Learning Latent Representations of 3D Human Pose with Deep Neural Networks. *International Journal of Computer Vision* 126, 12 (Jan 2018), 1326–1341. https://doi.org/10.1007/s11263-018-1066-6

[19] Kinect - Windows app development [n.d.]. https://developer.microsoft.com/en-us/windows/kinect/. (Accessed on 09/17/2020).

[20] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. 2020. VIBE: Video Inference for Human Body Pose and Shape Estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Jun 2020). https://doi.org/10.1109/cvpr42600.2020.00530

[21] Nikos Kolotouros, Georgios Pavlakos, Michael Black, and Kostas Daniilidis. 2019. Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the Loop. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct 2019). https://doi.org/10.1109/iccv.2019.00234

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs.CV]

[23] Charles Malleson, Andrew Gilbert, Matthew Trumble, John Collomosse, Adrian Hilton, and Marco Volino. 2017. Real-Time Full-Body Motion Capture from Video and IMUs. *2017 International Conference on 3D Vision (3DV)* (Oct 2017). https://doi.org/10.1109/3dv.2017.00058

[24] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. 2017. Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision. *2017 International Conference on 3D Vision (3DV)* (Oct 2017). https://doi.org/10.1109/3dv.2017.00064

[25] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. 2020. XNect: real-time multi-person 3D motion capture with a single RGB camera. *ACM Transactions on Graphics* 39, 4 (Jul 2020). https://doi.org/10.1145/3386569.3392410

[26] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017. VNect: real-time 3D human pose estimation with a single RGB camera. *ACM Transactions on Graphics* 36, 4 (jul 2017), 1–14. https://doi.org/10.1145/3072959.3073596

[27] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. 2019. 3D Human Pose Estimation With 2D Marginal Heatmaps. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. https://doi.org/10.1109/wacv.2019.00162

[28] OptiTrack - Motion Capture Systems [n.d.]. https://optitrack.com/. (Accessed on 09/17/2020).

[29] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Jun 2019). https://doi.org/10.1109/cvpr.2019.00794

[30] Thomas Pintaric and Hannes Kaufmann. 2008. A rigid-body target design methodology for optical pose-tracking systems. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology - VRST '08*. ACM Press. https://doi.org/10.1145/1450579.1450594

[31] Gerard Pons-Moll, Andreas Baak, Juergen Gall, Laura Leal-Taixe, Meinard Muller, Hans-Peter Seidel, and Bodo Rosenhahn. 2011. Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. *2011 International Conference on Computer Vision* (Nov 2011). https://doi.org/10.1109/iccv.2011.6126375

[32] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. 2019. Cross View Fusion for 3D Human Pose Estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. https://doi.org/10.1109/iccv.2019.00444

[33] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *arXiv* (2018).

[34] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. 2016. EgoCap. *ACM Transactions on Graphics* 35, 6 (Nov 2016), 1–11. https://doi.org/10.1145/2980179.2980235

[35] Martin Schepers, Matteo Giuberti, and G. Bellusci. 2018. Xsens MVN: Consistent Tracking of Human Motion Using Inertial Sensing. https://doi.org/10.13140/RG.2.2.22099.07205

[36] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. 2019. xR-EgoPose: Egocentric 3D Human Pose From an HMD Camera. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Oct 2019). https://doi.org/10.1109/iccv.2019.00782

[37] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. 2017. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. *Procedings of the British Machine Vision Conference 2017* (2017). https://doi.org/10.5244/c.31.14

[38] Vicon | Award Winning Motion Capture Systems [n.d.]. https://www.vicon.com/. (Accessed on 09/17/2020).

[39] Welcome to Steamworks [n.d.]. https://partner.steamgames.com/vrlicensing. (Accessed on 09/17/2020).

[40] Bin Xiao, Haiping Wu, and Yichen Wei. 2018. Simple Baselines for Human Pose Estimation and Tracking. *Lecture Notes in Computer Science* (2018), 472–487. https://doi.org/10.1007/978-3-030-01231-1_29

[41] Weipeng Xu, Avishek Chatterjee, Michael Zollhofer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. 2019. Mo2Cap2: Real-time Mobile 3D Motion Capture with a Cap-mounted Fisheye Camera. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (May 2019), 2093–2101. https://doi.org/10.1109/tvcg.2019.2898650

[42] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-Wall Human Pose Estimation Using Radio Signals. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. https://doi.org/10.1109/cvpr.2018.00768

[43] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 2018. 3D Human Pose Estimation in RGBD Images for Robotic Task Learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)* (May 2018). https://doi.org/10.1109/icra.2018.8462833

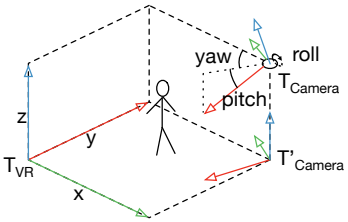

Figure 12: Transform between the camera and VR coordinates: T_{Camera} is the camera space, T'_{Camera} is the modified camera space, finally, T_{VR} is the VR space. *HybridTrak-transform* uses a neural network to generate 3D coordinate poses in the T'_{Camera} modified camera space, and then uses LSF to match the 3D poses to the T_{VR} space.

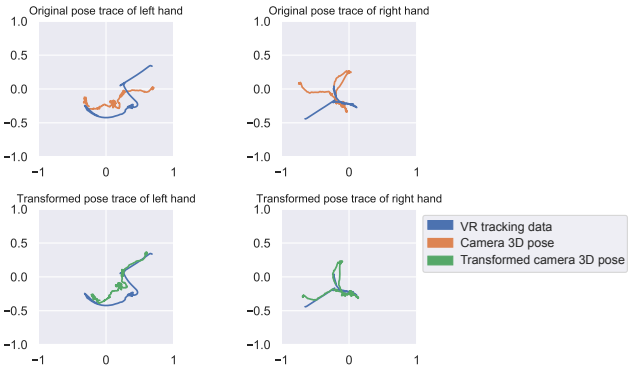

Figure 13: *HybridTrak-transform* uses a least-squares fitting to compute a rotation and scaling transform to align estimated 3D head and hand poses from the webcam with the head and hands positions from the VR tracking data. The original trace is on top, and the transformed trace is on the bottom. The transformed traces from the webcam are well aligned with the VR tracking traces.

A HYBRIDTRAK-TRANSFORM IMPLEMENTATION

We first use a neural network to process the 2D pose data from the webcam and generate six tracking points with positions and orientations. In the second step, we use "least-squares fitting"(LSF) to match the head and hands tracking points to the data from VR and produce a transformation between the camera space and the real-world space. In the end, we can apply that transformation to the predicted waist and feet poses to produce data for VR full-body tracking.

For the first step, we adopt a temporal convolutional neural network (CNN) similar to VideoPose3D [29]. We modify the network so that it outputs six tracking points with seven outputs, representing the position in three dimensions and the rotation in quaternion. Similar to regular HybridTrak, the network also accepts normalized 2D poses as input.

HybridTrak-transform does not require the camera extrinsic matrix to be supplied to the system so as to ensure a calibration-free

experience. The extrinsic matrix includes the camera's position (x, y, z) and orientation information (row, pitch, yaw). This information is typically used to transform the estimated 3D pose in the regular camera space to the VR space. To handle the position offset, we train the neural network to predict joint positions relative to the head. As we already know the head position, we can easily add the offset to the model to handle the position offset. To handle the camera rotation, we have to rely on both the neural network and our LSF algorithm. As the neural network is usually good at learning body geometry, and earth's gravity ensures that the users' center of gravity lies within their feet most of the time, we let the neural network directly produce the 3D body coordinates without pitch and yaw (see the modified camera space in Figure 12). With those coordinates, we only have to worry about the yaw differences between the modified camera space and the VR space.

Both the neural network and the VR tracking provide us with the user's hand position, so we can use this information to figure out the correct rotation between the two spaces. In other words, we need to generate a rotation to minimize the distance between the user's hand positions predicted by the neural network and the hand positions from the VR tracking system. In practice, we also noticed that the neural network may have some errors in estimating the size of the skeleton of the user, causing the user's leg positions to be further away or closer than where they should be. So we also take this chance to estimate a scale to re-scale the skeleton on the x-y plane. We generated this scaling and rotation transform using the least-squares fitting algorithm, an example result is shown in Figure 13.

So, with the appropriate rotation and scale transform from LSF, we can apply the transform to the result of the neural network and add the head position to all relative joints positions to get accurate positions and orientations in VR space.